Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

з виконаної лабораторної роботи № 6 (Пролог)

Дисципліна: Комп`ютерна логiка

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2018

**Лабараторна работа № 6**

Тема*:* “Робота зі списками на мові програмування Turbo Prolog”.

## Мета: Розглянути організацію спискових інформаційних структур та виконання основних операцій над ними на мові програмування Turbo Prolog.

***Завдання:***

Створити два списки: список прізвищ (наприклад своїх одногрупників) і список їх оцінок по будь-якому предмету. Представити у вигляді меню такі операції над цими списками:

1. Пощук заданого елемента в списку;
2. Поділ списку за заданим дільником;
3. Приєднання списків;
4. Сортування списку.

TASK\_1

domains

student\_group = student\*

mark\_list = mark\*

mark = integer

student=symbol

predicates

student\_list(student\_group)

student\_mark\_list(mark\_list)

% Отображение

display\_list(student\_group,mark\_list)

show

% Поиск

find(student,student\_group)

find\_in\_list

% Разделение

%split(integer,student\_group, student\_group,student\_group)

split(student,student\_group,student\_group,student\_group,mark\_list, mark\_list, mark\_list)

split\_list

% Обьединение

join(student\_group,student\_group,student\_group, mark\_list,mark\_list,mark\_list)

create\_list(student, student\_group,integer,mark\_list)

join\_list

% Сортировка

insert\_sort(student\_group,student\_group,mark\_list,mark\_list)

insert(student,student\_group,student\_group,mark,mark\_list,mark\_list)

ord(mark,mark)

sort\_list

main\_menu

task(integer)

execute

repeater

clauses

% списки по заданию

student\_list(["Annaev","Smirnov","Ivanov","Kovalenko","Koval"]).

student\_mark\_list([4,4,5,4,5]).

% Поиск студента в группе

% Запрос find\_in\_list

find(HEAD,[HEAD|\_]). % Сравнение головы условие выхода с рекурсии

find(HEAD,[\_|TAIL]):-find(HEAD,TAIL). % Выделение головы из хвоста

find\_in\_list:-

student\_list(STUDENT\_LIST),

write("Input student surname: "),nl,

readln(STUDENT\_SURNAME),find(STUDENT\_SURNAME,STUDENT\_LIST),

write("Present"),nl;

write("Not present"),nl.

% Разделение cписка оценок

split(X,[HEAD|TAIL],[HEAD|TAIL1],TAIL2,[M|TAIL3],[M|TAIL4],TAIL5):-HEAD<=X, split(X,TAIL,TAIL1,TAIL2,TAIL3,TAIL4,TAIL5).

split(X,[HEAD|TAIL],TAIL1,[HEAD|TAIL2],[M|TAIL3],TAIL4,[M|TAIL5]):-HEAD>X, split(X,TAIL,TAIL1,TAIL2,TAIL3,TAIL4,TAIL5).

split(\_,[],[],[],[],[],[]).

split\_list:-write("Input student surname: "),nl,

readln(STUDENT\_NAME),

student\_list(STUDENT\_LIST),student\_mark\_list(MARK\_LIST),

split(STUDENT\_NAME,STUDENT\_LIST,L1,L2,MARK\_LIST,M1,M2),write(L1),nl,write(M1),nl,write(L2),nl,write(M2).

% Отображение

display\_list([HEAD|TAIL],[HEAD1|TAIL1]):-write(HEAD, " ", HEAD1), nl, display\_list(TAIL,TAIL1).

show:-student\_list(STUDENT\_LIST),student\_mark\_list(MARK\_LIST) ,display\_list(STUDENT\_LIST,MARK\_LIST).

% Присоединение нового элемента в список студентов с его оценкой

join([],STUDENT\_LIST,STUDENT\_LIST,[], MARK\_LIST,MARK\_LIST).

join([STUDENT\_LIST|STUDENT\_LIST\_1],STUDENT\_LIST\_2,[STUDENT\_LIST|STUDENT\_LIST\_3],[MARK\_LIST|MARK\_LIST\_1],MARK\_LIST\_2,[MARK\_LIST|MARK\_LIST\_3]):-

join(STUDENT\_LIST\_1,STUDENT\_LIST\_2,STUDENT\_LIST\_3,MARK\_LIST\_1,MARK\_LIST\_2,MARK\_LIST\_3).

create\_list(STUDENT,[STUDENT|[]],MARK,[MARK|[]]).

join\_list:-

student\_list(STUDENT\_LIST),

student\_mark\_list(MARK\_LIST),

write("Input student surname: "),nl,

readln(STUDENT\_SURNAME),

write("Input student mark: "),nl,

readint(MARK),

create\_list(STUDENT\_SURNAME,NEW\_STUDENT\_LIST,MARK,NEW\_MARK\_LIST),

join(STUDENT\_LIST,NEW\_STUDENT\_LIST, TEMP\_1,MARK\_LIST,NEW\_MARK\_LIST,TEMP\_2),display\_list(TEMP\_1, TEMP\_2).

% Сортировка оценок

insert\_sort([],[],[],[]).

insert\_sort([X|TAIL\_1], SORTED\_LIST\_1, [Y|TAIL\_2], SORTED\_LIST\_2):-

insert\_sort(TAIL\_1,SORTED\_TAIL\_1,TAIL\_2,SORTED\_TAIL\_2),

insert(X,SORTED\_TAIL\_1,SORTED\_LIST\_1,Y,SORTED\_TAIL\_2,SORTED\_LIST\_2).

insert(X,[TAIL\_1|SORTED\_LIST\_1],[TAIL\_1|SORTED\_LIST\_TEMP1],Y,[TAIL\_2|SORTED\_LIST\_2],[TAIL\_2|SORTED\_LIST\_TEMP2]):-

ord(Y,TAIL\_2),!,insert(X,SORTED\_LIST\_1,SORTED\_LIST\_TEMP1,Y,SORTED\_LIST\_2,SORTED\_LIST\_TEMP2).

insert(X,SORTED\_LIST\_1,[X|SORTED\_LIST\_1],Y,SORTED\_LIST\_2,[Y|SORTED\_LIST\_2]).

ord(Y,TAIL\_2):-Y>TAIL\_2.

sort\_list:-student\_list(STUDENT\_LIST),student\_mark\_list(MARK\_LIST),insert\_sort(STUDENT\_LIST,STL,MARK\_LIST,ML),display\_list(STL,ML).

task(0):-exit.

task(1):-find\_in\_list.

task(2):-split\_list.

task(3):-join\_list.

task(4):-sort\_list.

task(5):-show.

main\_menu:-

write("0 - Exit"),nl,

write("1 - Find"),nl,

write("2 - Split"),nl,

write("3 - Join list"),nl,

write("4 - Sort list"),nl,

write("5 - Show list"),nl.

repeater.

repeater:-repeater.

execute:-repeater,main\_menu,readint(X),task(X),!.

goal

execute.

Контрольні питання:

1. Яким чином створити спискову структуру в Turbo Prolog?

domains

type = integer\*

predicates

list(type)

clauses

list ([1,2,3])

1. Що таке голова та хвіст списку?

Елемент списку, що самий перший – голова списку, а все інше – хвіст.

1. Як в Turbo Prolog здійснити поділ списку за заданим дільником?

domains

type = integer

list = integer\*

predicates

example\_predicates(m type list, list, list)

clauses

example\_predicates(X,[HEAD|TAIL],[HEAD|TAIL1],TAIL2):- HEAD <= X,

example\_predicates (X,TAIL,TAIL1,TAIL2).

example\_predicates (X,[HEAD| TAIL], TAIL1,[HEAD| TAIL2]): -

example\_predicates (X, TAIL, TAIL1, TAIL2), HEAD >X.

example\_predicates ( \_, [ ], [ ], [ ] ).

goal

example\_predicates (6, [3,7,8,6], X, Y).

1. Чи можна в Turbo Prolog здійснити введення списку з клавіатури?

Taк.